言語モデルにもとづいた機械翻訳実験 (計算機構論研究会報告集)

AUTHOR(S):

西村, 恕彦

CITATION:

西村, 恕彦. 言語モデルにもとづいた機械翻訳実験 (計算機構論研究会報告集). 数理解析研究所講究録 1969, 75: 71-90

ISSUE DATE:

1969-06

URL:

http://hdl.handle.net/2433/107970

数理解析研究所講究録
75巻 1969年 71-90

言語モデルにもとづいた機械翻訳実験

通商産業省 電気試験所
西村恕彦

1　概要

1・1　はじめに

筆者の作成した機械翻訳システムは，単に特定の言語を翻訳するだけのためのプログラムではない。一般的な記号の列を処理するためのプログラムである。

記号列の処理の手法は，いちいちの利用者が自分で記述しなければならない。その記述の形式は，メタ言語的な約束によって強く制限されている。逆にいえば，少数個のメタ言語的な約束だけにしたがって，容易に対象言語の性質を記述することができると考えられる。

ここで主として報告したいのは，いわゆる自然言語の翻訳を行なうために，メタ言語的な約束をどれほど設定しなければならなかったか，ということである。

1・2　構成

計算機の中には，辞書，文法表，処理プログラムの三者を入れておく。ここへ入力記号列を与えると，それにしたがって出力記号列が得られる。これが機械翻訳である。

辞書というのは，対象言語の性質をメタ言語的な約束にしたがって記述した規則の集合である。普通は，原語文字列，品詞，訳語文字列で一つの規則になる。

文法表というのは，対象言語の性質をメタ言語的な約束にしたがって記述した規則の集合である。普通は，品詞の列とそれで構成される何の品詞で一つの規則になる。

処理プログラムというのは，入力記号列を辞書や文法表の記述にしたがって，メタ言語的なしかたで解釈して変換し，出力記号列を作り出す算法である。

1・3　過程

翻訳は次のようにすすめられる。まず原語文字と呼ばれる記号の列，すなわち入力記号列$\mathfrak{X}$を，品詞と呼ばれる記号の列$\mathfrak{Y}_0$に書き換える。記号列$\mathfrak{Y}$は，書き換えが可能なあいだは重ねて何回も書き換えられ，記号列$\mathfrak{Y}_n$となる。最後にこの記号列$\mathfrak{Y}_n$が，訳語文字と呼ばれる記号の列，すなわち出力記号列$\mathfrak{Z}$に展開される。

記号列の書き換えの規則は，あるきまった値を有する記号列Rの「識別」と，それにたいする「処理」とからなっている。

記号列$\mathfrak{X}$を記号列$\mathfrak{Y}_0$に書き換える仕事は，文末品詞と呼ばれる特別の記号γに行き当たったときに中断され，$\mathfrak{Y}$の書き換えに移行する。

記号列$\mathfrak{Y}$の書き換えは，そのなかに識別できる記号列Rが

もはやなくなったときに中断され，記号列$\mathbb{Z}$の展開に移行する。

1．4　逐次処理

原文の要素の列（$\mathbb{X}$または$\mathbb{Y}$）と文法規則の集合とを照合して文法的な解析処理をすすめてゆくのに，おおざっぱにいって文走査型と逐次処理型とが区別できる。

文走査型では，文法規則の一つをとりあげて，それで文（始端，終端の限定された記号列）全体を端から端まで走査し，その規則があてはまる部分を処理する。これを，順序づけられた文法規則のすべてについてくりかえすものである。

逐次処理型では，文の端をとりあげて，文法規則の集合中にそれがあてはまる規則があるかどうかを走査する。文の端が処理できたら，（できなくても，）新しい端をとりあげてくりかえす。

筆者のシステムでは，まず文字列$\mathbb{X}$の処理を逐次処理で行ない，それが文の終端までゆくと，次に品詞列$\mathbb{Y}$の処理を逐次処理で行なう。どちらの場合にて照合は，逐次処理，見出しとの一致，最長一致，右方向の原則による。

文字列$\mathbb{X}$の処理は文の左から右へただ一回だけ行なわれる。このときの入力は原文の文字列$\mathbb{X}$で，出力は語の品詞列$\mathbb{Y}$。

である。ここで文のすがたは，文字の層から品詞（または語）の層にうつるのである。

品詞列𝕐の処理は文の左から右へ反復して行なわれる。品詞列𝕐に処理をくわえた結果はふたたび品詞列𝕐になる。だからその品詞列𝕐について重ねて文法規則との照合をとるのである。

文字列処理および品詞列処理の規則の集合はそれぞれ表の形になっている。その表の各項目はすべて，見出し（引数）と関数とからなっている。

1・5　要約

𝕏：　原文の文字列

𝕐：　品詞の列

ℤ：　訳文の文字列

R：　識別された記号列。きまった値を有し，長さは1以上

S：　処理（書き換え）される記号列。Rに含まれ，長さは0以上

T：　書き換える記号列

辞書の規則：　Rは𝕏である。Sは常にRに等しいものとされる。Tには二種類ある。第一種のTは，長さ1の品詞列𝕐と長さ0以上の訳語文字列ℤとからなっている。第二種のTは長さ1以上の原語文字列𝕏である。

文法表の規則：　Rは𝕐である。Sは𝕏に含まれ，長さは0以上である。Tの長さは，0，1，またはSの長さに等しい。Tの値は表面上は𝕐である。

普通品詞：　辞書中できまった値の𝕏にたいして与えられ，のちに訳語文字列ℤに展開される。

文末品詞：　普通の語の品詞とほとんど同じであるが，文字列処理段階を品詞列処理段階に切り替える仕きを，重ねて有する。

空品詞：　辞書中で，きまった値の𝕏にたいして与えられ，書き換えた結果のTは空となる。

未知語品詞：　二つの識別された文字列ではさまれた，長さ1以上の，識別されない一つの文字列𝕏にたいして，固有の品詞が与えられる。のちに展開される

文字列ℤの値は𝕏の値の前に1字の空白を付けたものに等しい。

文字列の書き換え：　辞書でこれが指定されると，識別された文字列　𝕏が，品詞𝕐ではなく，別の文字列𝕏に書き換えられる（第二種のT）。

句：　長さ1以上の𝕐を，長さ1の𝕐で書き換える。これには展開のしかたが幾通りかある。長さ1以上の𝕐を抹消することもできる。

文法語：　長さ0以上の𝕐を，長さ1の𝕐で書き換える。後者の𝕐は長さ0以上のℤを暗に含む。

2　文字列の処理

2.1　語の識別と有限状態文法

入力記号列𝕏の要素（単位）は文字である。文字の集合はホレリス文字と呼ばれる，英字，数字，特殊記号，空白を含む48種の文字からなる。それらのあいだには取扱い上，何の区別もない。如何なる類別も，対応もない。

原文の文字列𝕏は，文字列処理の規則にしたがって逐次処理される。文字列処理の規則の集合は辞書と呼ばれる。

辞書の見出しは原語文字列である。この文字列Rによって識別された文の構成要素Sを語という。辞書の関数はその語の品詞，およびその語を訳文に展開するための訳語文字列である。識別された文字列は語yに変換され，もとの文字列※は失なわれてしまう。

文字列※を語yに書き換える規則は，いわゆる有限状態文法にほぼしたがうものと考えられる。また本システムでは，文字，語，文という三種の単位しかみとめていないが，システムによっては，単語というような単位を設定し，それは，語幹およびそれにつながる0個以上の語尾からなるとすることもある。その場合の，語幹から語尾，語尾から語尾への連鎖の規則は，やはり有限状態文法にほぼしたがうとされている。

2・2　文字列の書き換え

文字列※は普通は語yに書き換えられるが，そうではなくふたたび文字列※に書き換えてもよい。これがなされたときには，もとの文字列は失なわれ，新しい文字列が文の文字列の一部となる。だからここでは逐次処理はいわば足踏み状態になる。

書き換えた文字列※について，重ねて何回でも書き換えが

おこってもよいが，いつかは語yに変換されなければならない。

文字列から文字列への書き換えはまったく任意でよい。したがってこの規則を利用して解析あるいは生成される言語は有限状態言語をこえていてよい。一般の自然言語処理をこの規則によって行なうこともできよう。

しかしその目的には具合のわるい制限もある。そのうちの四点をあげよう。

まず，層間規則との結びつきがわるいことである。つまり文字列の書き換えは真に書き換えであって，書き換えられたもとの文字列の値は失なわれ，どのような形でも保存されることはない。書き換えの経過があとで必要になっても利用はできず，文の解析につづいて生成をするとか，品詞情報を文字情報に変換するとかいった通常の翻訳には利用できない。

次に，かぎられた反復処理しかできないことがある。右方向の逐次処理で，ある文字列の書き換えが足踏みしてくりかえしているあいだはよいが，いったん処理が右へうつると，その左側の文字列は失なわれてしまい，あとからもう一度見直すことはできない。だから文字列の左端（処理端）は書き直す以外の方法では保存できない。

第三に，あるきまった文字列が識別されたときには一意の書き換えしかできない。だからきまった要素からいろいろな語を生成する，ランダムゼネレーションのような実験はできない。

第四に，文字列の識別，書き換えはその値を直接指定してしかできない。文字列中の一部が任意の値でよいとか，ある値を他の値で代入するとか，ある値を名前で呼ぶとかいうことができない。

筆者としては，この文字列の書き換えにそれほど大きな機能を予定していたわけではないので，うえのことは別に欠点ではない。つまり文字列の書き換えは，たとえば英和翻訳における，英語の不規則変化形の処理といったような，小さい仕事にだけつかうつもりにしている。

3　品詞列の処理

3・1　メタ言語

品詞列処理の規則は狭義の文法と呼ばれる。文の品詞列Yの左端の品詞列を文法表と照合する。もし照合がとれなけれ

ばひとつ右にずらした品詞列について照合する。

文の品詞列Yの中にある品詞列が，文法表の見出しと一致したとき，それを識別された品詞列Rという。識別された品詞列R中の一つの部分列Sが下で書き換えられる。書き換えがすむと，ふたたび文の左端にもどって次の照合がくりかえされる。

識別された記号列R中の部分列Sを，他の記号列下で書き換える，この書き換えはまったく任意なのではなく，記号列の長さや，展開規則との結び付きによって，強く制限されている。

記号列Yの中で記号列Rが識別されたときの処理の，メタ言語的な約束には七種類ある。そのうち六種類では記号列Sの書き換えがおこり，そのほかの一種類では書き換えはおこらない。

（1） 句構造の代入

Sの長さは1以上であって，下の長さは1である。下は暗にSの情報をそのまま含み，それがのちに展開される。

a b c d e f → a b m f

（２） 逆順の句構造の代入

Sの長さは２以上であって，Tの長さは１である。表面上，Yに関しては（１）の規則と同じである。Tは暗に記号列Sの情報を逆順に保存し，それがのちに展開される。

$$abcdef \rightarrow abmf$$

（３） 語の代入

Sの長さは１以上であって，Tの長さは１である。表面上は（１）の規則と同じである。Tは記号列Sの情報を抹消し，かわりに固有の値Zを暗に保有する。

$$abcdef \rightarrow abmf$$

（４） 語の挿入

Sの長さは０であって，Tの長さは１である。つまり，識別された記号列R中の，指定したある位置に，一つの記号yが割り込む。yは固有の値Zを暗に保有する。

$$abcdef \rightarrow abncdef$$

（５） 抹消

Sの長さは１以上であって，Tの長さは０である。つまり，識別された記号列R中の部分列Sを抹消する。抹消された値はどのようにしても，回復されることもないし，展開されることもない。

a b c d e f → a b f

（６） 転置

Sの長さは２以上である。Tの長さはSの長さに等しく，その値は，記号列Sの構成要素を逆順にならべたものに等しい。

a b c d e f → a b e d c f

（７） 飛び越し

記号列Sの書き換えはおこらない。うえの（１）から（６）までの場合には，書き換えがすむと，記号列Yと文法表との照合は，記号列Yの左端にもどって再開される。

しかし，飛び越しの場合には，記号列Rが識別されると，次の照合は記号列Y中のSの位置から始められる。

いわゆる結合の強さの優先順位を処理する都合のために，

この指定を開発した。

3・2　評価

うえに述べた七種類の規則を，メタ言語の立場から評価してみよう。

規則（１）と（２）とは，あきらかに句構造文法，それも見かけ上は，文脈検査型の句構造文法である。おそらく（３）代入と（７）飛び越しともまたそれに属するのであろう。

規則（６）転置は，いわゆる変形規則を満足させるために導入したものである。実際に利用してみると，転置がおこつたことをしるしづけるものがＹの中に必要なことがあり，それで規則（４）挿入と関係づけて用いることが多い。

規則（５）抹消の意義はあきらかではないが，（３）代入，（４）挿入と関係があるらしい。

規則（４）挿入で気付いたことであるが，この機能はもつと拡大することができる。この規則で挿入することができるのは，固有の値ｙ，Ｚを有する語だけである。それをたとえば識別された記号列Ｅ中の任意の記号（または記号列）の値Ｙ，Ｚを「複写」できるように拡張すると，このシステムの能力は格段に大きくなる。それによつてたとえば，記号微分，記号計算（括弧をはずす），鏡像記号列の生成といったよ

うなことが可能になる。

現在のシステムでは，それらは一般的には可能ではない。

識別される記号列Rや書き換える記号列Tの指定は，きまった値についてだけ可能である。それを記号列中の一部の記号が任意の値でよいというようにすることも考えられ，SNOBOLやCOMITでは普通の装置となっている。

その応用例としては，たとえば冠詞と名詞とのあいだに何がはさまっていても，全体として名詞句になるとか，接続詞の両側に同じ品詞の句があったときには，全体としても同じその品詞の句になるとかがある。

集合の要素としての記号は，相互にまったく独立で，固有の値を有するものとしてある。実際の言語事象では，大文字と小文字とか，清音と濁音とかのあいだには，あきらかにある種の関係がある。あるいは動詞の細分類というような例もある。記号について，集合の集合とか，部分集合とかいうとらえ方を導入することも考えられる。

そのほかとくに英語に関係した話題として，一致（呼応）の処理，多義語，多品詞語，多材能品詞などがある。

4　文字列の展開

品詞列𝕐から文字列𝕑への展開は，左から右へただ1回だけ行なわれる。訳文の文字列は語を構成する文字からなり，語と語とのあいだでは単純な連結だけがなされ，それ以外の文字の抹消，挿入，変更などは何もなされない。

未知語の直前に空白が1字入る以外には，語と語とのあいだにはどのような区切り文字も挿入されない。

語yの文字列𝕑は，語順𝕐中のその位置に展開されるだけで，そのほかの位置には影響しない。

訳語の語順は品詞処理における品詞の語順処理のすべてを，そしてそれだけを反映している。品詞処理のそれぞれが語順におよぼす影響はほとんど自明であるが，あらためて以下に述べる。

品詞yには語の品詞と句の品詞とがある。語の品詞には一つの語の文字列𝕑が直接に従属している。語の品詞および句の品詞の，1以上の長さの列に代入したのが句の品詞である。つまり句の品詞には句または語の品詞の列𝕐が直接に従属している。

文字列の展開にあたって，各品詞の値は何の意義ももたない。そして文字列𝕑中に含められることもない。すなわち，

語の文字列はどのような品詞の下に従属していてもまったく同じである。そして訳文は，語の文字列と未知語を区切る空白とだけからなる。

（１） 句構造の代入と展開語順

句構造の代入が行なわれると，その句の中の語順は固定され，それ以後変更することはできない。句の中の語順は正方向の展開か逆方向の展開かのいずれかだけを選択できる。だから長さnの品詞列にはnの順列通りの順序が可能であるが，それが一つの句構造として代入されるときには，二通りの語順のいずれかしか許されない。

（２） 逆順の句構造

句の中の語順を逆方向と指定したとき，その効果は，その句を直接に構成する品詞列についてだけ有効である。すなわち，句を構成する品詞がふたたび句の品詞であった場合に，語順の指定は下位の句同士のあいだの順序はさだめるが，下位の句の内部の語順を変更することはけっしてない。

（３） 語の代入

ある品詞列にたいして語を代入すると，その品詞列に従属す

る語の文字列はすべて抹消される。そしてそのかわりに，ただ一つの語の文字列がその位置を占める。抹消された語が回復することはけっしてない。

（4） 語の挿入

語を挿入するとその位置にその語の文字列が入る。前後の語には，文字位置以外の何も影響をおよぼさない。

（5） 抹消

ある品詞列を抹消すると，それに従属する語の文字列はすべて抹消される。抹消されたことをしるしづけるものは何も残らない。抹消された語はどのような形でも回復することはない。

（6） 転置

ある品詞列Sを転置すると，それに従属する語や句もまた転置される。転置はそれらの語や句のあいだの語順を逆にするが，それより下位の句の内部の語順を変更することはけっしてない。

転置がおこったことをしるしづけるものは何もないし，転置された品詞列の一部または全部がふたたび転置その他あら

ゆる処理をくわえられてかまわない。

（７）　飛び越し

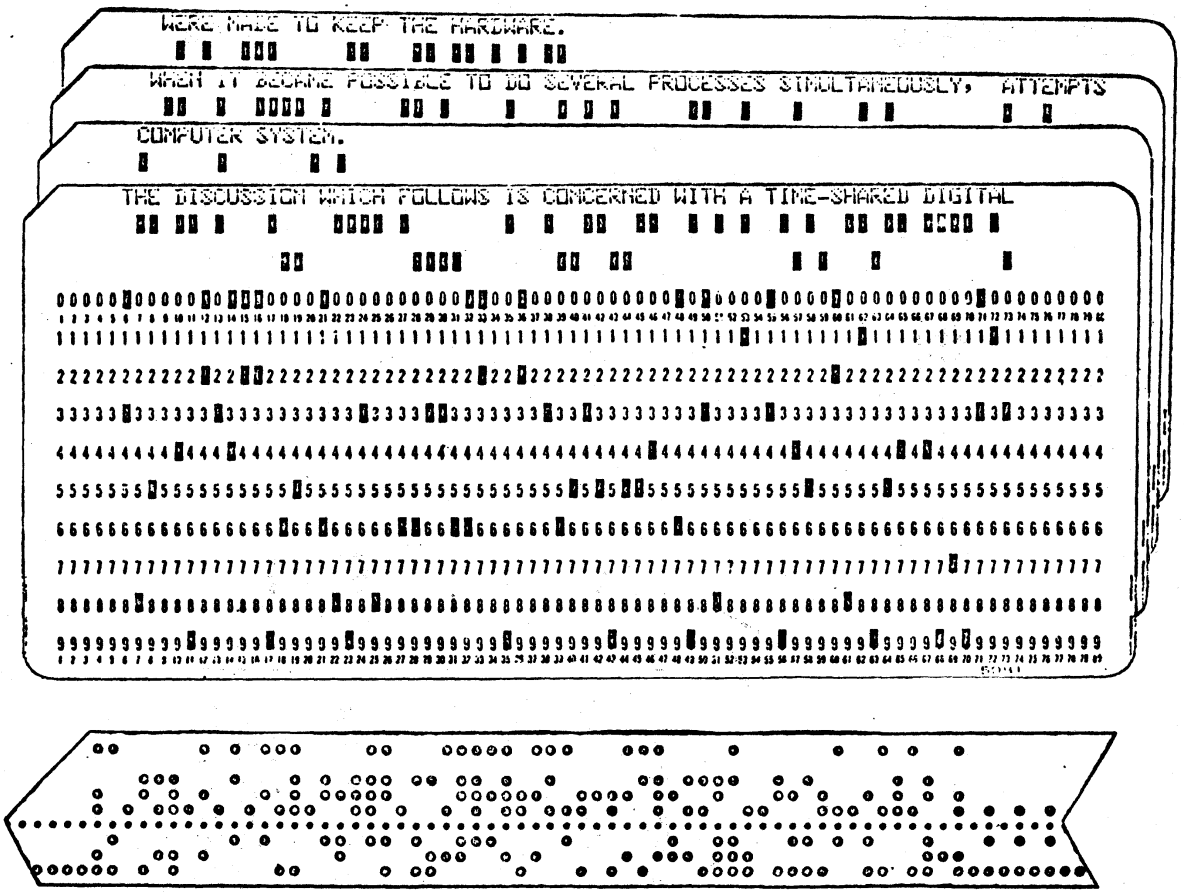
飛び越しは品詞列の照合に関係するだけで，文字列の展開には何も影響しない。

5　　例

R	T	S
形　形	形	０２代
名	句	０／代
形　名	句	０２代
句　名	句	０２代
句　前　句	句	０３逆
句　前　句　名	句	２２代
句　前　形		２△飛
句　前　名	句	２／代

The discussion which follows is concerned with a time-shared digital computer system.

When it became possible to do several processes simultaneously (usually, but not necessarily, computation and input-output) attempts were made to keep the hardware (in particular, the central processor) as busy as possible.



後に続くところの議論は時分割計数型計算機組織に関係される。

同時に幾つかの過程をなすことが可能となった時、試みはハードウェアを保つために作られた。